

**Amendments to the Claims:**

This listing will replace all prior versions, and listings, of claims in the application.

**Listing of Claims:**

---

1-10. (cancelled).

11. (currently amended) A method, comprising:

1.  
B  
a first process detecting one or more script language instructions in a markup language document, wherein the first process is implemented in a platform-independent programming language;

generating an intermediate representation of the one or more script language instructions, wherein the intermediate representation of the one or more script language instructions is different from the script language; and

interpreting and executing each of one or more instructions of the intermediate representation, wherein said interpreting and executing accesses one or more program objects to implement the one or more instructions of the intermediate representation;

wherein said interpreting and executing produces results in accordance with the original one or more script language instructions.

12. (previously presented) The method as recited in claim 11, wherein the first process is a Web browser, the method further comprising:

the Web browser passing execution to an interpreter engine after said detecting the one or more script language instructions in the markup language document;

wherein said generating, said interpreting and executing, and said accessing one or more program objects are performed by the interpreter engine.

13. (previously presented) The method as recited in claim 11, wherein the one or more instructions of the intermediate representation are stored in a stack data structure, and wherein said interpreting and executing is performed by a stack-machine interpreter engine that pops the one or more instructions off the stack data structure during said interpreting and executing.

14. (previously presented) The method as recited in claim 11, wherein said generating an intermediate representation of the one or more script language instructions comprises validating that the one or more script language instructions conform to script language syntax.

15. (previously presented) The method as recited in claim 11, wherein said generating an intermediate representation of the one or more script language instructions comprises modifying a first of the one or more program objects.

16. (previously presented) The method as recited in claim 15, wherein said modifying the first program object comprises adding at least one of methods and fields to the first program object.

17. (previously presented) The method as recited in claim 15, wherein said modifying the first program object comprises deleting at least one of methods and fields from the first program object.

18. (previously presented) The method as recited in claim 11, wherein the one or more program objects are accessed from one or more object libraries.

19. (previously presented) The method as recited in claim 11, wherein the script language is JavaScript.

20. (previously presented) The method as recited in claim 11, wherein the markup language is Hypertext Markup Language (HTML).

21. (previously presented) The method as recited in claim 11, wherein each of the one or more instructions of the intermediate representation is represented as one or more Java objects each referencing one of the one or more program objects accessed in implementing the one or more instructions of the intermediate representation.

22. (previously presented) The method as recited in claim 21, wherein each of the one or more Java objects of the intermediate representation represents one of an operator and an operand of the particular instruction of the intermediate representation.

23. (previously presented) The method as recited in claim 11, wherein the one or more program objects are Java objects.

24. (previously presented) The method as recited in claim 11, wherein the platform-independent programming language is the Java programming language.

25. (previously presented) The method as recited in claim 11, wherein the first process is a Java-based Web browser executing within a Java Virtual Machine.

26. (previously presented) A method, comprising:

a Web browser implemented in a platform-independent programming language examining a current tag of a markup language document marking the beginning of a portion of the markup language document;

if said examining determines the current tag of the markup language document identifies the portion of the markup language document as comprising script language instructions, passing execution to an interpreter engine implemented in the platform-independent programming language; and

if said examining determines the current tag of the markup language document does not identify the portion of the markup language document as comprising script language instructions, the Web browser executing the portion of the markup language document identified by the current tag;

wherein the interpreter engine is configured to perform the script language instructions to produce results in response to said passing execution.

27. (previously presented) The method as recited in claim 26, wherein the interpreter engine executing the script language instructions comprises:

generating an intermediate representation of the script language instructions; and

interpreting and executing each of one or more instructions of the intermediate representation, wherein said interpreting and executing accesses one or more program objects from one or more object libraries to implement the one or more instructions of the intermediate representation;

wherein said interpreting and executing produces results in accordance with the original script language instructions.

28. (previously presented) The method as recited in claim 27, wherein each of the one or more instructions of the intermediate representation is represented as one or more Java objects each referencing one of the one or more program objects accessed in performing the one or more instructions of the intermediate representation.

29. (previously presented) The method as recited in claim 27, wherein the one or more program objects are Java objects.

30. (previously presented) The method as recited in claim 26, wherein the script language is JavaScript.

31. (previously presented) The method as recited in claim 26, wherein the markup language is Hypertext Markup Language (HTML).

32. (previously presented) The method as recited in claim 26, wherein the platform-independent programming language is the Java programming language.

33. (previously presented) The method as recited in claim 26, wherein the Web browser and the interpreter engine are executing within a Java Virtual Machine.

34. (currently amended) A device, comprising:

a processor;

a memory coupled to said processor and operable to store program instructions in a platform-independent programming language, wherein the program instructions are executable by the processor to:

detect one or more script language instructions in a markup language document;

generate an intermediate representation of the one or more script language instructions, wherein the intermediate representation of the one or more script language instructions is different from the script language; and

interpret and execute each of one or more instructions of the intermediate representation to produce results in accordance with the original one or more script language instructions;

wherein, in said interpreting and executing, the program instructions are further executable by the processor to access one or more program objects to implement the one or more instructions of the intermediate representation.

35. (previously presented) The device as recited in claim 34, wherein the program instructions implement a Web browser and an interpreter engine, wherein the Web browser is executable by the processor to perform said detecting one or more script language instructions, and wherein the Web browser is further executable by the processor to pass execution to the interpreter engine after said detecting, and wherein the interpreter engine is executable by the processor to perform said generating, said interpreting and executing, and said accessing one or more program objects.

36. (previously presented) The device as recited in claim 35 wherein the program instructions further implement a Java Virtual Machine executable by the processor within the device, and wherein the Web browser is a Java-based Web browser executable within the Java Virtual Machine.

37. (previously presented) The device as recited in claim 34, wherein, in said generating an intermediate representation of the one or more script language instructions, the program instructions are further executable by the processor to modify a first of the one or more program objects.

38. (previously presented) The device as recited in claim 37, wherein, in said modifying a first of the one or more program objects, the program instructions are further executable by the processor to add at least one of methods and fields to the first program object.

39. (previously presented) The device as recited in claim 37, wherein, in said modifying a first of the one or more program objects, the program instructions are further executable by the processor to delete at least one of methods and fields from the first program object.

40. (previously presented) The device as recited in claim 34, wherein, in said generating an intermediate representation of the one or more script language instructions, the program instructions are further executable by the processor to validate that the one or more script language instructions conform to script language syntax.

41. (previously presented) The device as recited in claim 34, wherein the one or more program objects are accessed from one or more object libraries.

42. (previously presented) The device as recited in claim 41, wherein the memory is further operable to store the one or more object libraries.

43. (previously presented) The device as recited in claim 34, wherein the script language is JavaScript.


44. (previously presented) The device as recited in claim 34, wherein the markup language is Hypertext Markup Language (HTML).

45. (previously presented) The device as recited in claim 34, wherein each of the one or more instructions of the intermediate representation is represented as one or more Java objects each referencing one of the one or more program objects accessed in implementing the one or more instructions of the intermediate representation.

46. (previously presented) The device as recited in claim 34, wherein the one or more program objects are Java programming language objects.

47. (previously presented) The device as recited in claim 34, wherein the platform-independent programming language is the Java programming language.

48. (previously presented) A device, comprising:



a processor;

a memory coupled to said processor and operable to store program instructions in a platform-independent programming language, wherein the program instructions implement a Web browser and an interpreter engine;

wherein the Web browser is executable by the processor to:

process a markup language document comprising a script comprising one or more script language instructions;

detect the script in the markup language document; and

provide the script to the interpreter engine;

wherein the interpreter engine is executable by the processor to execute the script to produce results of the script.

49. (previously presented) The device as recited in claim 48, wherein the platform-independent programming language is Java, wherein the program instructions further implement a Java Virtual Machine executable by the processor within the device, and wherein the Web browser and the interpreter engine are executable within the Java Virtual Machine.



50. (previously presented) The device as recited in claim 48, wherein, in said executing the script, the interpreter engine is further executable by the processor to:

generate an intermediate representation of the script; and

execute the intermediate representation to produce results in accordance with the original script.

51. (previously presented) The device as recited in claim 50, wherein, in said executing the intermediate representation, the interpreter engine is further executable by the processor to interpret and execute each of one or more instructions of the intermediate representation.

52. (previously presented) The device as recited in claim 50, wherein, in said executing the intermediate representation, the interpreter engine is further executable by the processor to access one or more program objects from one or more object libraries to implement one or more instructions of the intermediate representation.

53. (previously presented) The device as recited in claim 48, wherein each of the one or more instructions of the intermediate representation is represented as one or more Java objects each referencing one of the one or more program objects accessed in implementing the one or more instructions of the intermediate representation.

54. (previously presented) The device as recited in claim 48, wherein the script language is JavaScript.

55. (previously presented) The device as recited in claim 48, wherein, in said detecting the script, the Web browser is further executable by the processor to detect a markup language tag identifying the start of the script.

56. (previously presented) A system, comprising:

a device;

a Java Virtual Machine executable within the device;

an interpreter engine executable within the device; and

a Web browser executable within the Java Virtual Machine to:

detect one or more script language instructions in a markup language document; and

pass execution to the interpreter engine in response to said detecting;

wherein the interpreter engine is executable within the device to:

generate an intermediate representation of the detected one or more script language instructions; and

interpret and execute each of one or more instructions of the intermediate representation to produce results in accordance with the original one or more script language instructions;

wherein, in said interpreting and executing, the interpreter engine is further executable within the device to access one or more program objects to implement the one or more instructions of the intermediate representation.

57. (previously presented) The system as recited in claim 56, wherein, in said generating an intermediate representation of the one or more script language instructions, the

interpreter engine is further executable to modify a first of the one or more program objects.

58. (previously presented) The system as recited in claim 56, wherein the one or more program objects are accessed from one or more object libraries.

59. (previously presented) The system as recited in claim 56, wherein the script language is JavaScript.

60. (previously presented) The system as recited in claim 56, wherein each of the one or more instructions of the intermediate representation is represented as one or more Java objects each referencing one of the one or more program objects accessed in implementing the one or more instructions of the intermediate representation.

61. (previously presented) The system as recited in claim 56, wherein the one or more program objects are Java programming language objects.

62. (currently amended) A carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement:

a first process detecting one or more script language instructions in a markup language document, wherein the first process is implemented in a platform-independent programming language;

generating an intermediate representation of the one or more script language instructions, wherein the intermediate representation of the one or more script language instructions is different from the script language; and

interpreting and executing each of one or more instructions of the intermediate representation, wherein said interpreting and executing accesses one or

more program objects to implement the one or more instructions of the intermediate representation;

wherein said interpreting and executing produces results in accordance with the original one or more script language instructions.

63. (previously presented) The carrier medium as recited in claim 62, wherein the first process is a web browser, wherein the program instructions are further computer-executable to implement:

the Web browser passing execution to an interpreter engine after said detecting the one or more script language instructions in the markup language document;

wherein said generating, said interpreting and executing, and said accessing one or more program objects are performed by the interpreter engine.

64. (previously presented) The carrier medium as recited in claim 62, wherein, in said generating an intermediate representation of the one or more script language instructions, the program instructions are further computer-executable to implement modifying a first of the one or more program objects.

65. (previously presented) The carrier medium as recited in claim 62, wherein the script language is JavaScript.

66. (previously presented) The carrier medium as recited in claim 62, wherein the one or more program objects are Java programming language objects, and wherein each of the one or more instructions of the intermediate representation is represented as one or more Java objects each referencing one of the one or more program objects accessed in implementing the one or more instructions of the intermediate representation.

B1  
ms

67. (previously presented) The carrier medium as recited in claim 62, wherein the first process is a Java-based Web browser executing within a Java Virtual Machine.

---